Day 9	**Logic Gates**	**6-12-2015** **7 -12-2015**

In this chapter you will learn about:

- **Logic gates**
- **Truth tables**
- **Logic circuits/networks**

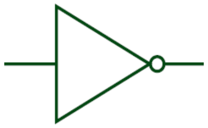
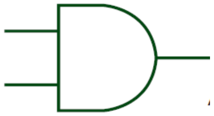
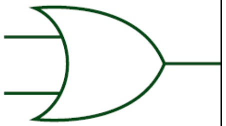
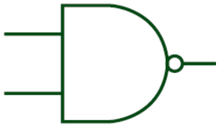
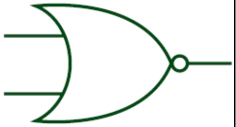
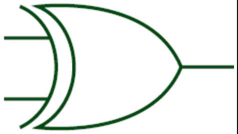
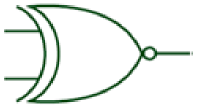
In this chapter we will look at how logic gates are used and how truth tables are used to check if combinations of logic gates (known as logic circuits or logic networks) carry out the required functions.

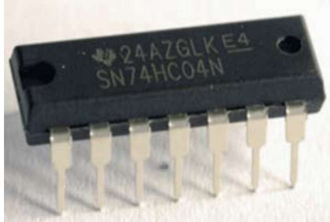
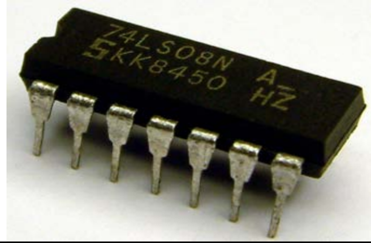
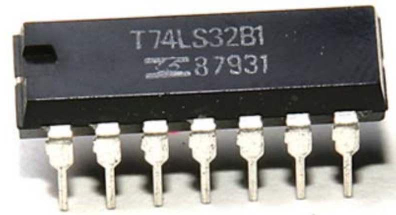
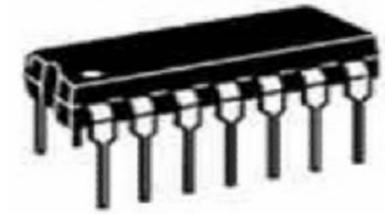
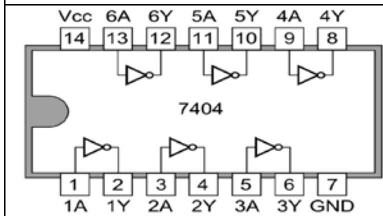
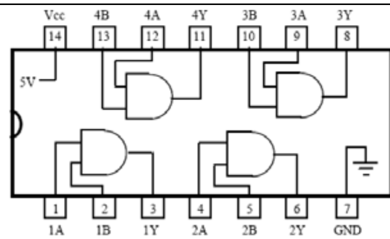
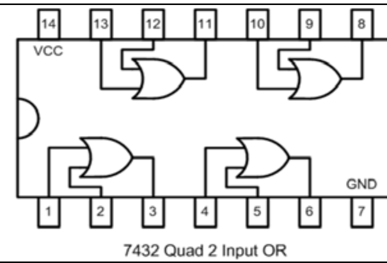
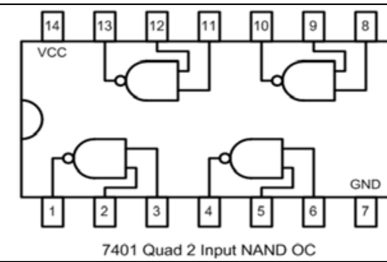
● Logic gates

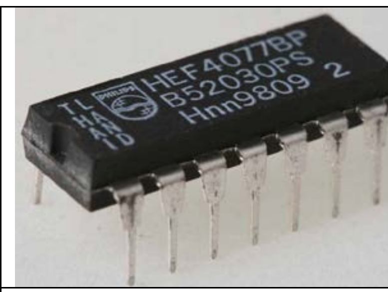
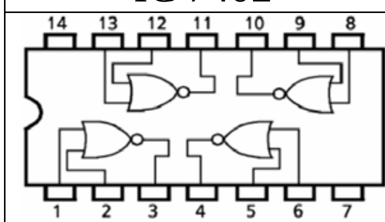
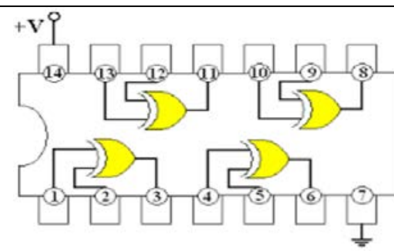
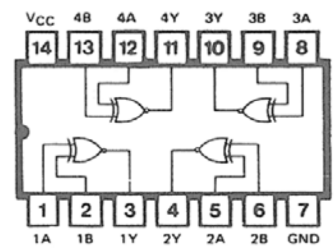
- are electric circuit called gates.
- the electronic circuits in computers and control units are made up of these gates.
- Logical values can easily be expressed by an electric signal:

  “True” or “1” can be defined as voltage on a wire .... Signal is ON

  “False” or “0” can be defined as No voltage on a wire... Signal is OFF
- There are many different logic gates but we will concentrate on these.

NOT gate	AND gate	OR gate	NAND gate	NOR gate	XOR gate	XNOR gate
						

			
NOT gate	AND gate	OR gate	NAND gate
IC 7404	IC 7408	IC 7432	IC 7401
			

		
NOR gate	XOR gate	XNOR gate
IC 7402	IC 7486	IC 4077
		

● Truth tables

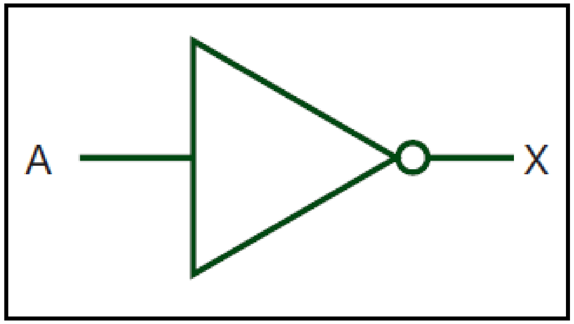
◇ Truth tables are used to show logic gate functions. The NOT gate has only one input, but all the others have two inputs. When constructing a truth table, the binary values 1 and 0 are used. Every possible combination is produced depending on number of inputs.

◇ Basically; the number of possible combinations of 1's and 0's is 2^n where n = number of inputs.

For example, 2 inputs have 2^2 combinations (i.e. 4),

3 inputs have 2^3 combinations (i.e. 8)

NOT gate = Inverter



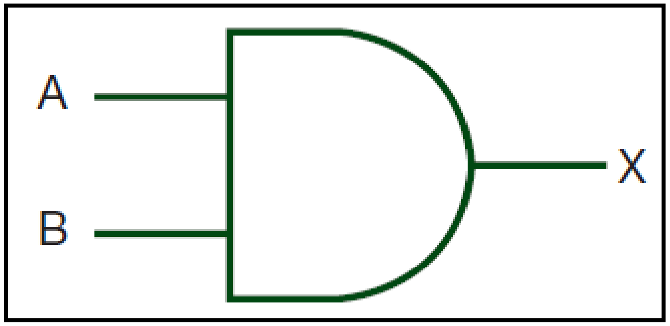
INPUT A	OUTPUT X
0	1
1	0

The output (X) is **true** if **INPUT A** is **NOT TRUE**

- The NOT gate is an electronic circuit that produces an output from an inverted version of the input.
- If the input variable is A, the inverted output is known as NOT A.
- This is also shown as A', or Ā with a bar over the top

Truth table for: ***X = NOT A = A’ =*** $\boldsymbol{\overline{A}}$

AND gate



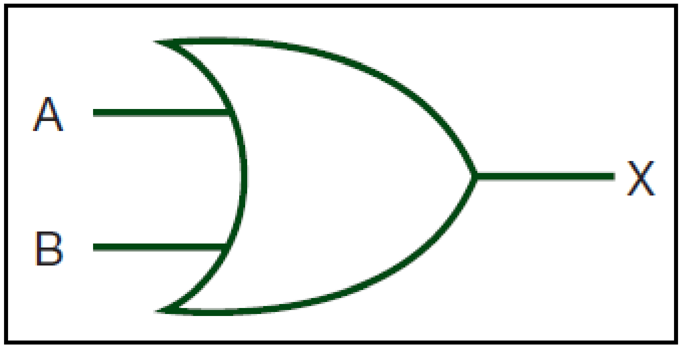
INPUT A	INPUT B	OUTPUT X
0	0	0
0	1	0
1	0	0
1	1	1

The output (X) is **true** if **INPUT A AND INPUT B** are **BOTH TRUE**

• The AND gate is an electronic circuit that gives a true output (1) only if all its inputs are true.
• A dot (·) is used to show the AND operation i.e. A·B. Note that the dot is sometimes omitted i.e. AB

Truth table for: ***X = A AND B = A.B = AB***

OR gate



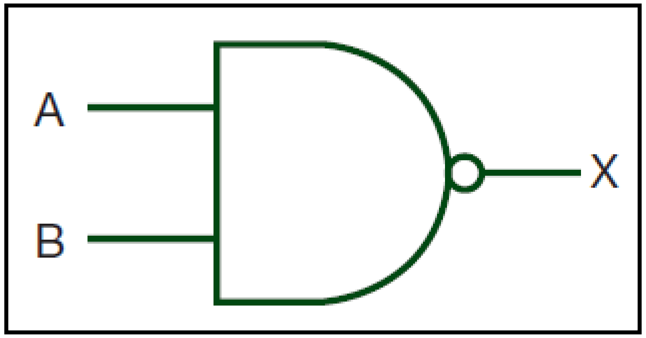
INPUT A	INPUT B	OUTPUT X
0	0	0
0	1	1
1	0	1
1	1	1

The output (X) is **true** if **INPUT A OR INPUT B** is **TRUE**

• The OR gate is an electronic circuit that gives a true output (1) if one or more of its inputs are true. A plus (+) is used to show the OR operation.

Truth table for: ***X = A OR B = A+B***

NAND gate



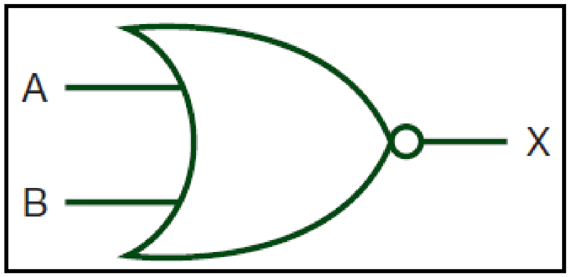
INPUT A	INPUT B	OUTPUT X
0	0	1
0	1	1
1	0	1
1	1	0

The output (X) is **true** if **INPUT A AND INPUT B** are **NOT BOTH TRUE**

- This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate.
- The outputs of all NAND gates are true if any of the inputs are false.
- The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

Truth table for: ***X = NOT A AND B*** $= \overline{A.B} = \overline{AB}$

NOR gate

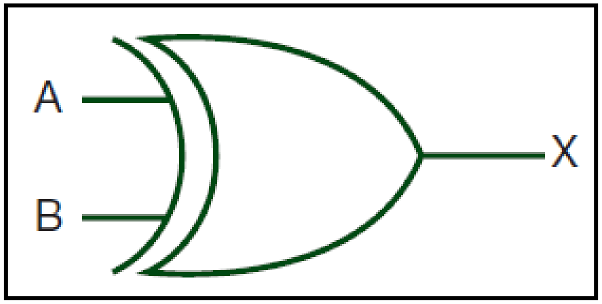


INPUT A	INPUT B	OUTPUT X
0	0	1
0	1	0
1	0	0
1	1	0

- This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate.
- The outputs of all NOR gates are false if any of the inputs are true.
- The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

Truth table for: ***X = NOT A OR B*** = $\overline{A+B}$

XOR gate

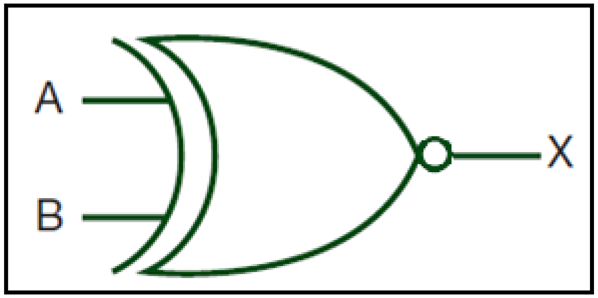


INPUT A	INPUT B	OUTPUT X
0	0	0
0	1	1
1	0	1
1	1	0

- The 'Exclusive-OR' gate is a circuit which will give a true output if either, but not both, of its two inputs are true.
- An encircled plus sign (⊕) is used to show the XOR operation.

Truth table for: ***X = A OR (NOT B) = (NOT A) OR B = A ⊕ B***

XNOR gate



INPUT A	INPUT B	OUTPUT X
0	0	1
0	1	0
1	0	0
1	1	1

- The 'Exclusive-NOR' gate circuit does the opposite to the XOR gate.
- It will give a FALSE output if either of inputs is true.

Truth table for: $X = \overline{A \oplus B}$

Logic circuits/networks

- Logic gates can be combined together to produce more complex logic circuits (networks).

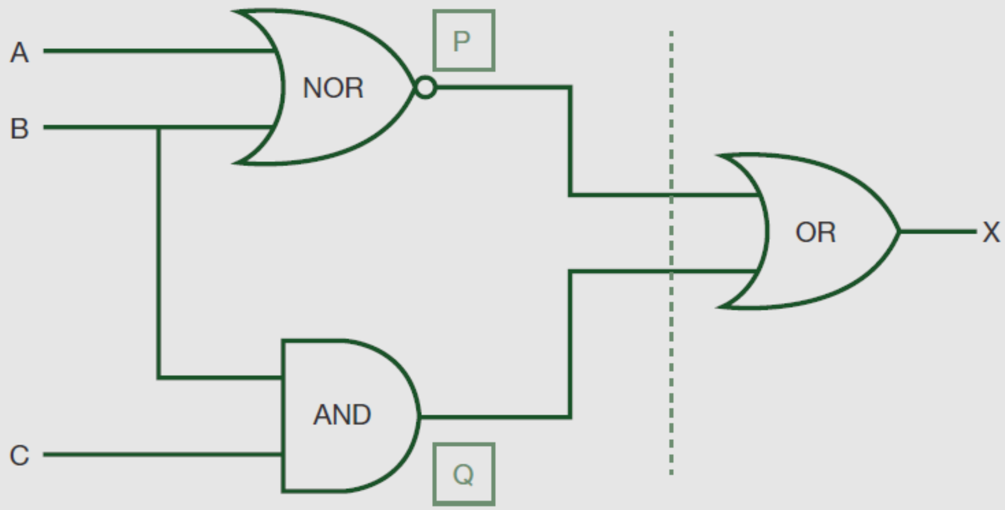
Two different types of problem are considered here:

– drawing the truth table from a given logic circuit (network)

– designing a logic circuit (network) from a given problem and testing it by also drawing a truth table.

Example 1

Produce a truth table from the following logic circuit (network).



Answer

There are 3 inputs; thus we must have 2^3 (i.e. 8) possible combinations of 1s and 0s.

A	B	C	P	Q	X
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

Example 2

A system used 3 switches A, B and C; a combination of switches determines whether an alarm, X, sounds: If switch A or switch B are in the ON position and if switch C is in the OFF position then a signal to sound an alarm, X is produced.
Convert this problem into a logic statement.

Answer:

So we get:

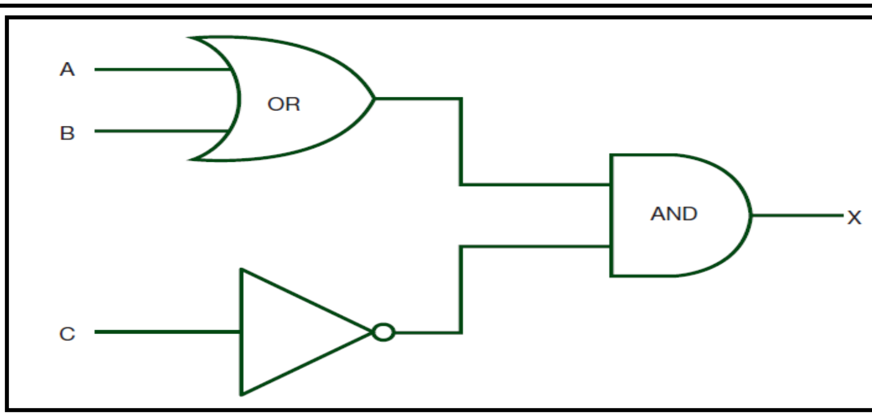
If (A = 1 **OR** B = 1) **AND** (C = **NOT** 1) then X = 1

The first part is two inputs (A and B) joined by an **OR** gate

The output from the first part and the third part are joined by an **AND** gate

The third part is one input (C) which is put through a **NOT** gate

INPUT A	INPUT B	INPUT C	OUTPUT X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



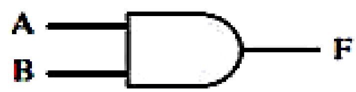
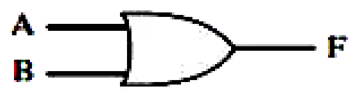
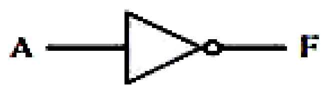
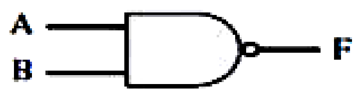
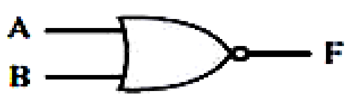
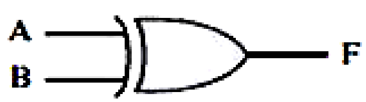
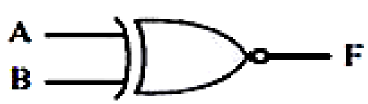
Example 3

Using the truth table of basic gates to show that:

$$\overline{A \bullet B} = \overline{A} + \overline{B}$$

ANSWER:

A	B	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$	$A \bullet B$	$\overline{A \bullet B}$
0	0	1	1	1	0	1
0	1	1	0	1	0	1
1	0	0	1	1	0	1
1	1	0	0	0	1	0

Name	Graphical Symbol	Algebraic Function	Truth Table
AND		$F = A \bullet B$ or $F = AB$	A B | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = A + B$	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
NOT		$F = \overline{A}$ or $F = A'$	A | F 0 | 1 1 | 0
NAND		$F = \overline{AB}$	A B | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = \overline{A + B}$	A B | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
XOR		$F = A \oplus B$	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
XNOR		$F = \overline{A \oplus B}$	A B | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1